



PROJET PPII  
PROJET PLURIDISCIPLINAIRE D'INFORMATIQUE INTÉGRATIVE

Rapport de projet :  
Démocratie Participative

*Auteurs :*

Thomas KIEFFER

Julien DE TOFFOLI

Nathan IORI-GINGEMBRE

Pierre PASQUIER



# Table des matières

<b>Introduction</b>	<b>1</b>
<b>1 Etat de l’art</b>	<b>2</b>
1.1 La démocratie participative . . . . .	2
1.2 La démocratie participative en parallèle avec internet : les « Civic Tech » . . . . .	2
1.3 Les applications de la Civic Tech en ligne . . . . .	3
1.3.1 Différentes caractéristiques au profit d’une participation renforcée . . . . .	3
1.3.2 Caractéristiques souhaitées pour l’application cible . . . . .	5
<b>2 Conception et implémentation des algorithmes</b>	<b>6</b>
2.1 threeBestInterest : suggérer par la consultation . . . . .	6
<b>3 Conception et implémentation des Bases de Données</b>	<b>7</b>
3.1 Tables Demandes et Offres . . . . .	7
3.2 Tables Postulants et Demandeurs . . . . .	8
3.3 Table Utilisateurs . . . . .	8
3.4 Table Notation . . . . .	9
3.5 Tables Mairie, Sondage et isSondage . . . . .	9
3.5.1 Mairie . . . . .	9
3.5.2 Sondage . . . . .	9
3.5.3 isSondage . . . . .	10
<b>4 Conception et implémentation des pages Web</b>	<b>11</b>
4.1 Création d’annonces, (titre à améliorer) . . . . .	11
4.2 Mes demandes, mes offres et leur postulants . . . . .	11

4.3	13
4.4	13
<b>5 Tests et performances de l'application</b>	<b>14</b>
<b>Gestion de Projet</b>	<b>15</b>
5.1 Forces et faiblesses de notre projet : matrice SWOT	15
5.2 Gestion des risques : évaluation de la fréquence et de la gravité des événements	16
5.3 Découpage des lots de travail	17
5.4 Répartition des responsabilités	18
<b>Conclusion</b>	<b>19</b>
<b>Annexes</b>	<b>19</b>
<b>Annexe 1 : Comptes rendus</b>	<b>20</b>
<b>Annexe 2 : Diagramme de Gantt</b>	<b>31</b>
<b>Annexe 3 : Gestion des risques</b>	<b>32</b>



# Introduction

Plus le temps passe, moins les citoyens on l'impression d'avoir réellement le pouvoir au sein d'une démocratie représentative, et plus ils se détachent des autres concitoyens. Il faudrait donc trouver un moyen de redonner envie à ces concitoyens de participer a la vie en société. C'est l'origine de la démocratie participative.

Dans ce cadre nous est donné un projet à réaliser sur deux mois à quatre.

Nous allons chercher à créer une application qui va nous permettre mettre en place la démocratie participative au sein d'une ville.

Nous sommes donc partis du principe que notre application faciliterait les échanges entre les mairies et les citoyens afin que leurs pouvoir démocratique puisse s'exprimer de façon plus simple et qu'elle leur permette d'avoir une plus grande capacité d'action sur l'avenir de leurs propre ville.

Pour ce faire nous allons donc mettre en place un système de sondage qui permettra aux mairies d'avoir l'avis de ses concitoyens de façon plus simple que par la mise en place de vote ou de prise de décisions qui nécessiteraient un déplacement physique. De plus l'application permettra d'améliorer l'échange entre les citoyens par le biais d'offres et de demandes qui faciliterait la vie de tout à chacun. Nous allons donc mettre en place des comptes spécifiques pour les mairies qui seraient identifiées comme tel et des comptes pour les citoyens .

# Chapitre 1

## Etat de l'art

### 1.1 La démocratie participative

La démocratie participative est un concept de démocratie où le citoyen prend plus part aux décisions politiques. Plus précisément, le public peut prendre part aussi bien à l'élaboration de ces décisions qu'à des processus permettant de les influencer. Ainsi, dans cet esprit de participation, le citoyen se détache de la démocratie représentative, en laquelle il n'a parfois plus confiance, comme en témoigne la participation au premier tour des législatives en France qui n'était que de 48.7 % en 2017.

Dans le sens de cette insuffisance de la démocratie représentative, la participation du citoyen dans la vie politique se popularise de plus en plus, notamment dans des applications qu'on catégorise comme « Civic Tech ».

### 1.2 La démocratie participative en parallèle avec internet : les « Civic Tech »

Comme le définit la Knight Foundation en 2013, les « Civic Tech » représentent n'importe quelle technologie **visant à accroître le pouvoir du citoyen ou à rendre un gouvernement plus ouvert**. Avec l'expansion d'internet, il est désormais plus facile pour un grand nombre de particuliers de s'exprimer, et de ce fait, les interactions entre les citoyens, au même titre que les citoyens eux-mêmes, constituent la clé de voûte de la démocratie. Donner la possibilité au citoyen de s'exprimer, c'est lui donner du pouvoir, donc Internet permet l'implémentation de Civic Tech, ce sur

quoi nous centrerons notre projet.

## 1.3 Les applications de la Civic Tech en ligne

Nous nous concentrerons en particulier sur les applications en ligne de la Civic Tech.

### 1.3.1 Différentes caractéristiques au profit d'une participation renforcée

Les applications qui ont pu être mises en place avec les années diffèrent de par leurs principes et leur public visé. En effet, nous avons répertorié différentes applications de démocratie participative dans un tableau, et en fonction de critères que nous avons établis, nous l'avons rempli, ce qui nous a mené à une catégorisation de ces applications. Parmi ces critères, on retrouve :

- - Public visé : Tout le monde n'est pas nécessairement visé avec la création d'une application
- - But : Même si c'est indirect, l'application peut viser quelque chose tout en impactant positivement la démocratie participative
- - Financement participatif : Les applications peuvent ou non traiter d'argent
- - Mise en relation des citoyens : Les citoyens sont au coeur de la démocratie, donc la façon de les faire communiquer avec une application peut s'avérer intéressant
- - Permet d'informer : L'information est le quatrième pouvoir, informer les gens c'est aussi les faire participer, donc il est intéressant de savoir si une application le permet
- - Système de vote : Le vote est un outil au service de la démocratie depuis ses débuts. Est ce que l'application est dotée d'une telle caractéristique ?
- - Moyen d'identification : Si les citoyens peuvent être mis en contact, un moyen de les identifier peut s'avérer utile. Comment est ce que les applications existantes procèdent ?
- - Proposition de services : Est ce que l'application permet aux citoyens de s'aider les uns les autres ? Si tel est le cas, comment est ce que l'application le permet ?
- - Donner son opinion sur différents sujets : Les citoyens ont-ils la possibilité de s'exprimer sur différents sujets ?

Nous avons donc utilisé ce tableau pour essayer de se demander : quelles cases notre application devrait-elle remplir ? Et comment les remplirait-on ? Parmi toutes les applications que nous avons répertoriées dans le fichier Excel, voici celles qui nous permettent de bien différencier les types d'applications de la Civic Tech :



TABLE 1.1 – Tableau de comparaison entre application de la Civic Tech

Applications	Public visé	But	Financement participatif	Relations	Information	Vote	Login	Proposition de service	Opinion
Tell my city	Citoyens d'une même ville	Participer au bien être de la ville	Non	Lancement d'alertes, débats publics	Sur les incidents, actualité de la commune	Non	Nom, email, tel	Non	Pour vérifier si les incidents ont bien eu lieu, commentaires des incidents
KKBB	Toute la population	Financement	Oui	Non	Sur les différents projets en cours	Non	Nom, email, tel	Oui par les financements	Non
Voisin-age	Personnes âgées de +50 ans	Contribuer au bien-être des aînés	Non	Réseau social	Non	Non	Nom, email, tel, adresse	prévention et lien social vers les personnes âgées isolées	Non
Nos députés.fr	Citoyens / Députés	Favoriser l'échange entre les députés et les citoyens	Non	Participer à des débats à propos de l'activité des députés à l'AN	Sur l'activité des députés	Non	(Plateforme en ligne)	Non	A propos de l'activité parlementaire

### **1.3.2 Caractéristiques souhaitées pour l'application cible**

Comme nous avons pu le voir, les applications de la Civic Tech représentent soit une application pour la ville avec un système d'alertes et de sondages (Tell my city), soit une application d'entraide (Voisin-âge), soit une application basée sur les échanges (Facebook) ou bien une application centrée sur l'information (Nos députés.fr). Notre application souhaitée rassemble donc les échanges, avec systèmes d'annonces, et les sondages pour les Mairies, pour demander l'avis des citoyens dans les missions qu'elles pourraient leur donner. A l'instar des autres applications, nous voudrions implémenter un système de login pour différencier les utilisateurs, et un système de cryptage de mot de passe.

# Chapitre 2

## Conception et implémentation des algorithmes

### 2.1 threeBestInterest : suggérer par la consultation

Notre algorithme threeBestInterest est l'élément clé permettant de pouvoir améliorer l'affichage des annonces pour les utilisateurs, permettant un système de suggestion par un score composé de nombre premier.

Afin de suggérer les filtres selon ce que visitent les utilisateurs, nous devons appliquer une métrique, en retenant les visites des utilisateurs. Pour faciliter ceci, nous avons utilisé un nombre premier par filtre, afin de multiplier le score\_demande et score\_offre de l'utilisateur lorsqu'il regarde une annonce associée, par le nombre premier du filtre en question. Ainsi son score représente un nombre décomposable en nombres premiers.

# Chapitre 3

## Conception et implémentation des Bases de Données

### 3.1 Tables Demandes et Offres

Concernant nos bases de données, nous avons commencé par nécessiter une table pour stocker nos demandes et nos offres. Ces tables sont très similaires, et les explications les concernant seront donc communes. Pour créer ces tables, on a besoin d'un moyen d'identifier chaque demande donc avec un `id_dem` comme clé primaire. On a ensuite besoin de `id_auteur_dem` pour pouvoir relier la demande à la personne qui l'a demandé. L'attribut `etat` permet de classer notre demande selon qu'elle soit en attente de réalisation et qu'il reste encore une place donc ' en attente ' , ' en cours ' si un candidat a été sélectionné par l'auteur de l'annonce, ' terminé ' dans le cas où la demande a été réalisée et clôturée par son auteur. L'attribut `id_accepteur` permet de référencer l'utilisateur étant accepté pour réaliser l'annonce. Les attributs `lieu` et `description` fournissent le lieu et la description plus détaillée sur la demande. L'attribut `Datedeb` quant à lui permet de savoir à quel moment la demande a été créée alors que `dateservice` permet de donner la date de réalisation souhaitée du service. Le `Titre` va permettre de donner une idée de ce sur quoi la demande va porter et les filtres permettent aux postulants de se retrouver beaucoup plus facilement entre les demandes.

## 3.2 Tables Postulants et Demandeurs

Dans les tables postulants et demandeurs, nous avons souhaités permettre d'associer un id de poste, l'id du postulant et son message. Cela a permis dans son utilisation d'à chaque nouveau postulant d'ajouter une ligne avec l'id du poste, du postulant et son message. Nous avons donc dû mettre l'id du poste et celui du demandeur en clé primaire. car l'annonce peut avoir plusieurs postulants, et les postulants peuvent postuler à plusieurs annonces.

## 3.3 Table Utilisateurs

Cependant pour pouvoir avoir des postulants et des demandeurs, il nous faut donc des utilisateurs, que nous stockerons dans une table Utilisateurs. L'attribut `id_utilisateur` est la clés primaire de la table et permet d'identifier les différents utilisateurs. L'attribut `score_offre` permet d'enregistrer les thèmes préférés des utilisateurs pour les offres et ainsi de leurs afficher des offres qui seraient plus susceptible de leurs correspondre, grâce à l'algorithme `threeBestInterest` vu plus haut dans implémentation et conception des algorithmes. L'attribut `score_demande` marche selon le même principe. Le `pseudo` est unique et permet à l'utilisateur d'avoir un compte sans forcément divulguer son identité en ligne. Le `mail` correspond au mail de l'utilisateur, il permet dans les cas où il est nécessaire de joindre l'utilisateur de façon extérieur à l'application, voire de vérifier son compte lors de sa création à l'aide d'un code. Il en est de même pour l'attribut `numTel` qui correspond au numéro de téléphone de l'utilisateur, bien qu'il ne soit pas particulièrement utilisé comme le mail. L'attribut `mdp` contient le mot de passe de l'utilisateur et permet donc aux différents utilisateurs d'avoir un compte privé bien sécurisé. La `description` permet d'écrire dans le profil quel type de citoyen l'utilisateur semble être.

## 3.4 Table Notation

Notre site utilise le principe de notation pour motiver à aider les utilisateurs, ainsi il faut une table Notation pour garder une trace de tout ça. On commence l'attribut `id` qui est la clé primaire de la table Notation, correspondant à l'id de l'utilisateur. L'attribut `note` va prendre une valeur comprise entre 1 et 5 avec 1 pour moins bonne note et 5 pour meilleure note et qui est le résultat de la moyenne de toutes les notes qu'il a obtenu par d'autres utilisateurs. L'attribut `nb_note` donne le nombre total de notes que l'utilisateur a déjà obtenu, permettant de calculer la moyenne précédente, et l'attribut `note_par` donne l'id des utilisateurs ayant donné une note à l'utilisateur possédant l'id de la table Notation, afin de ne pas permettre à quelqu'un d'être noté deux fois par la même personne.

## 3.5 Tables Mairie, Sondage et isSondage

### 3.5.1 Mairie

Pour bien inclure les mairies et les distinguer des utilisateurs, nous avons créé une table Mairie, table comprenant simplement `id_utilisateur`, non nul ainsi qu'un `varchar` `True` ou `False`, permettant avec la clé primaire `id_utilisateur` de pouvoir savoir si un utilisateur est une mairie ou non, afin de lui donner accès ou non à la création de sondages.

### 3.5.2 Sondage

Afin de permettre aux mairies de mieux permettre aux citoyens de participer à la vie de leur ville, elles auront exclusivement accès aux sondages, avec la table Sondages, qui comprend `id` qui est l'id de l'auteur, `type` qui correspond à offre ou demande, afin de pouvoir savoir à quoi est associé le sondage, pour que avec `typeid` l'id du poste, il n'y ai pas de confusion en donnant l'accès à la demande `i` et l'offre `i`. Ainsi, nous avons la clé primaire `id_sondage`, qui correspond à l'id du sondage pour le récupérer facilement, un attribut `questions`, comportant un `varchar` contenant les questions du sondage, de même pour les attributs `reponses` et `type_rep`, dont le dernier contient les éléments permettant de savoir si une question sera à choix multiple ou non.

### 3.5.3 isSondage

Cette table isSondage permet, en lien avec les mairies et les sondages, de vérifier si une annonce possède déjà ou non un sondage pour garder un unique sondage par poste, avec l'aide de type provenant de la table Sondages pour savoir de quel type d'annonce provient le sondage, l'id\_annonce lui aussi provenant de Sondages pour avoir avec type le poste exact du sondage, un varchar Bool, étant soit True soit False, afin de savoir si l'annonce possède déjà un sondage ou non pour ne pas lui en créer à nouveau, et enfin le varchar repondu\_par, contenant tout les utilisateurs ayant déjà répondu au sondage.

# Chapitre 4

## Conception et implémentation des pages Web

### 4.1 Création d'annonces, (titre à améliorer)

### 4.2 Mes demandes, mes offres et leur postulants

Concernant la partie plus personnelle pour l'utilisateur, afficher "mes demandes" et "mes offres" de l'utilisateur, bien que les deux parties sont fortement semblables, il a fallut faire des pages et des fonctions distinctes.

Tout d'abord a été prévu de faire les pages mes demandes et mes offres, pour cela il suffisait à l'aide d'une requête sql de récupérer les valeurs et à l'aide d'un for, de toute les afficher. Afin de les afficher de manière plus esthétique, nous avons utilisé un tableau, et transformé le titre de l'annonce dans la ligne en lien vers la page de l'annonce en particulier.

Il aurait été appréciable niveau esthétique que la ligne entière soit cliquable pour mener vers le lien, cependant cela demandait des ressources en css, html et javascript bien trop importante niveau jours homme.

Cela a permis au début de bien prendre en main le html et pouvoir retenir les bonnes méthodes de code. Cela a aussi permis niveau design de pouvoir se mettre à la place de l'utilisateur et de donc penser à mettre un lien vers créer un sondage, si jamais l'utilisateur sur sa page vide souhaitait la remplir.

Ensuite, concernant le poste en particulier, il suffisait tout d'abord d'afficher toute les infos de



la page dessus, cependant l'url comportait l'id de l'utilisateur, et donc il a fallut penser à vérifier si le poste en question appartenait à l'utilisateur lors de sa visualisation dans ses annonces. Nous avons ajouté niveau design un lien permettant de retourner vers toute les annonces de la personne. Niveau code web, cela a requit de faire passer par la plupart des pages l'id de l'utilisateur en paramètre afin de pouvoir rester connecter entre les déplacements sur le site. Une fois la page terminée, il a fallut prendre en compte les utilisateurs intéressés par l'annonce, qui y ont donc postulé, ainsi l'ajout du lien pour voir les personnes intéressés. Pour cela il a suffit de récupérer les informations des utilisateurs ayant postulé, avec encore une fois une boucle for suivant tout les postulants. Il a été possible d'y ajouter un lien permettant de revenir vers la demande en question, ainsi qu'ensuite un lien vers la selection d'un postulant afin de l'associer à la demande.

Cette fois, l'utilisation du for a permis de remarquer la possibilité de créer des boutons dans une boucle, et donc ici avec la sélection d'un postulant parmi ceux intéressés, de créer un bouton radio pour chaque postulant avec un for. Nous avons encore une fois ajouté la possibilité de revenir à la page précédente avec un lien en tête. Cependant cette page aura apporté quelques difficultés au niveau du code de l'architecture web de cette page, il a fallut distinguer la methode d'accès, GET et POST, dans lesquels le premier affichait la page normalement, et le second récupérait donc les valeurs des boutons radio, afin de pouvoir récupérer le postulant choisi. Il a aussi fallut ajouter des vérifications à ces méthodes, si aucun utilisateur n'est choisi, si l'annonce était bien en attente, autrement il n'est plus possible d'y postuler, et si un postulant n'a pas déjà été choisi. Qu'un postulant ai déjà été choisi ou qu'il vient d'être choisi, la page affichait le nom de l'utilisateur qui a été choisi, avec un bouton permettant de revenir vers la liste des postulants.

Pour en revenir à la page affichant l'annonce, une fois le postulant sélectionné, cela a permis de changer l'état de la demande a "en cours", ne l'affichant plus aux autres utilisateurs. Malheureusement, il n'a pas été possible niveau temps de faire une page mes annonces sélectionnées, bien qu'essentielle, cette fonctionnalité a été écartée par priorisation du déplacement fonctionnel sur le site ainsi que les autres fonctionnalités. Afin de continuer autour de l'état de la demande, nous avons ajouté deux boutons supplémentaires. Le premier permet de clôturer l'annonce, et donc de passer son état à "terminée" lorsque l'auteur aura décidé que le contenu autour de son annonce a été complété, et le deuxième, de supprimer la demande lorsque son auteur l'aura décidé. Ces deux fonctionnalités auront requit de vérifier que l'id de l'utilisateur tentant d'accéder à cette page est bien celui de l'auteur de la page, ainsi que de rediriger l'auteur vers ses annonces afin de ne pas le

laisser retourner sur la page désormais inexistante dans le cas de suppression.

En suivant la progression de l'écriture du code, nous avons décidé d'associer nos sondages aux annonces, et donc sur la page de l'annonce particulière, il a fallut ajouter un bouton pour créer un sondage associé à l'annonce. Ceci demandait de vérifier trois choses particulières concernant l'utilisateur et la demande. Tout d'abord, vérifier si l'utilisateur est une mairie, ensuite si le poste est en attente, et enfin si le poste ne comporte pas déjà de sondage associé. Si ces trois conditions sont suivies, le bouton de création est affiché, autrement non, et si l'utilisateur est tout de même une mairie mais que le poste a déjà un sondage, un simple message rappellera à la place du bouton qu'un sondage existe déjà.

Enfin, il a été réfléchi de permettre à la clôture d'un poste de rediriger vers la notation de l'utilisateur sélectionné. Il a donc fallut vérifier le cas où le poste n'avait pas de postulant sélectionné, auquel cas l'auteur est redirigé en cloturant vers ses demandes, au lieu d'être redirigé vers la page de notation de l'utilisateur.

## 4.3

## 4.4

## Chapitre 5

### Tests et performances de l'application

# Gestion de Projet

## 5.1 Forces et faiblesses de notre projet : matrice SWOT

Après analyse des forces et faiblesses, internes et externes de notre projet, nous en avons déduit notre matrice SWOT :



FIGURE 5.1 – Matrice SWOT

Ainsi, pour la gestion de projet, il s'agissait de bien prendre en compte les menaces et les faiblesses inhérentes au projet. En particulier, pour l'éloignement lors des vacances, nous nous sommes organisés afin de pouvoir tenir des réunions sur Discord. De plus les partiels ont pu être en partie anticipés en mettant au clair les conceptions des différentes parties (Web, Base de données, Algorithme). Quand bien même ces parties ont été sujettes à des modifications ultérieures (en particulier la base de données), la conception a permis de bien comprendre les besoins de l'application.

Par ailleurs, la possibilité de demander de l'aide aux 2A/3A nous a fortement aidés, en particulier pour les bases de données et le latex.

Les faiblesses de notre projet nous ont fait prendre conscience qu’une autoformation était nécessaire pour éviter certains problèmes, notamment les requêtes git merge

La force de notre projet, c’est à dire la cohésion, a permis au projet d’avancer, sans conflits. Les réunions et la communication, malgré certains problèmes, ont permis de mettre rapidement les choses au clair et de résoudre les problèmes assez rapidement. Nous n’avons pas souvent eu affaire à ‘l’effet tunnel’.

## 5.2 Gestion des risques : évaluation de la fréquence et de la gravité des événements

Les faiblesses du projet nous ont permis d’établir le diagramme de gestion des risques, en estimant les fréquences et les gravités :

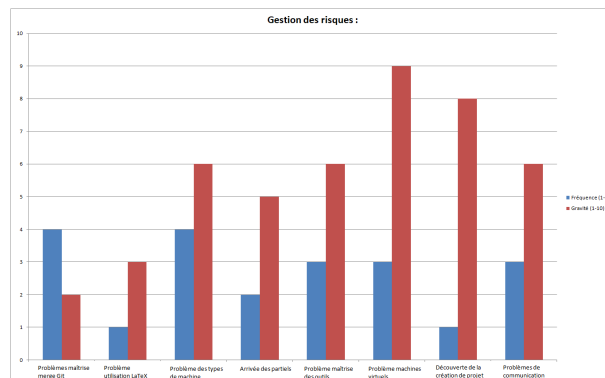


FIGURE 5.2 – Diagramme de fréquence et de gravité pour des risques éventuels

Certains problèmes n’ont pas pu être évités (cf Annexe 3 pour plus de visibilité), comme par exemple les problèmes des types de machines, c’est à dire la machine virtuelle de certains qui ne marchait pas à l’instant t. Mais cette analyse nous a permis, en estimant les fréquences et les gravités, de prendre conscience de ce qui pouvait être amélioré, comme la maîtrise des outils

## 5.3 Découpage des lots de travail

Suite à cette évaluation, nous avons construit le WBS :

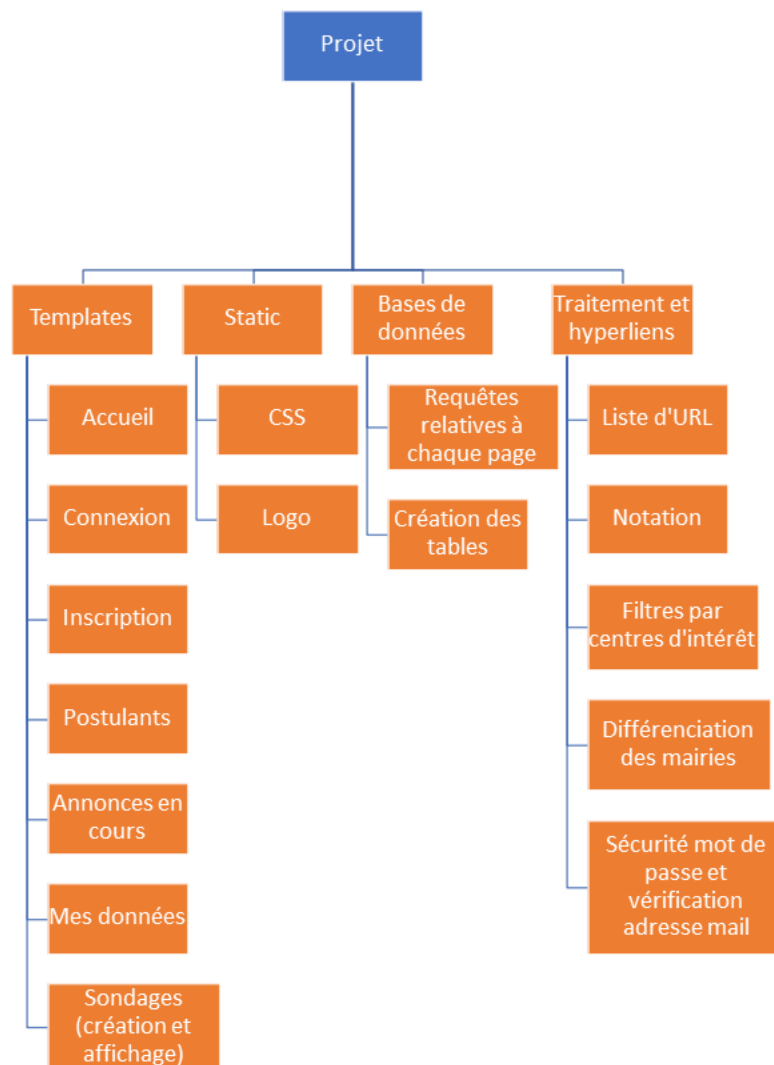


FIGURE 5.3 – Division du projet en lots de travaux

## 5.4 Répartition des responsabilités

Suite à cette division, nous avons établi la répartition des tâches :

	Julien	Nathan	Thomas	Pierre
Logo	R/A	C	C	I
CSS	A	C	R	I
Page de création de compte	C	I	A	R
Pages demandes et annonces en cours	C	I	A	R
Page mes demandes/mes offres	R	I	A	C
Page intéressés par les offres	R	I	A	C
Page création de sondage	C	I	R	A
Page affichage de sondage	C	I	R	A
Page intéressés par les demandes	C	I	R	A
Page mon compte	A	R	I	C
Page mon profil	A	R	I	C
Page création d'annonce	A	R	I	C
Gestion des URL	I	C	A	R
Notation des utilisateurs	I	C	A	R
Traitement des centres d'intérêts des utilisateurs	A	C/I	R	R
Tri par date	I	R	A	C
Référencement des mairies	R	A	C	I

TABLE 5.1 – Matrice RACI

# Conclusion

A travers notre projet nous avons pu créer une application de démocratie participative, en créant des sondages, des annonces ainsi que les interactions avec les utilisateurs et en incitant à participer à la vie communautaire à travers notre site. Cependant avec les difficultés de découverte de la gestion de projet, ainsi que notre projet un peu chargé, nous ne sommes pas parvenus à compléter tout ce que nous attendions de notre projet. En revanche, ce projet a été une expérience enrichissante qui a permis de développer nos compétences en web ainsi que notre travail en équipe. Il nous a aussi permis de voir sur le terrain la gestion de projet et la difficulté de sa mise en oeuvre, et nous permettra par la suite de ne pas refaire les mêmes erreurs et de travailler plus efficacement.



# Annexe 1 : Comptes Rendus

## Compte rendu 6 Novembre 2019

Nous sommes le 6 Novembre 2021, Julien, Pierre, Nathan et Thomas sont présents. L'objectif de cette réunion est de discuter sur les applications du MOOC dans le cadre de notre projet et de réfléchir à la forme de l'état de l'art pour répondre au sujet. La réunion a lieu sur Discord à 17h.

### *Répartition au sein du groupe*

- - Tout le monde développe
- - Nathan et Pierre seront testeurs
- - Julien sera responsable design
- - Thomas sera chef de projet

### *Décisions principales*

- - Faire des recherches sur les différentes applications de la Civic Tech
- - Classifier les différentes applications de la Civic Tech vis à vis des caractéristiques qui s'en dégagent

### *To Do List*

- - Description : Compléter le tableau de synthèse
- - Responsable : Tout le monde
- - Délai : pour la prochaine réunion le 11/11/2021

— - Validé par : Tout le monde

## ***Prochaine Réunion : 11/11/2021***

- - Compléter le tableau synthèse de l'état de l'art
- - En tirer un concept d'application

## Compte rendu 12 Novembre 2021

Nous sommes le 12 Novembre 2021, Julien, Pierre, Nathan et Thomas sont présents. L'objectif de cette réunion est de discuter du concept d'application, vis à vis de notre état de l'art. La réunion a lieu sur Discord à 17h.

### *Décisions principales*

- - Choix de l'orientation de l'application : annonces pour les citoyens et sondages pour les mairies pour compléter leurs annonces et impliquer les citoyens dans ses actions.
- - Consensus sur quelques besoins en bases de données relatifs aux fonctionnalités à programmer
- - Décision sur la page d'accueil du site internet à créer

### *To Do List*

- - Description : Demander des conseils à Mme Heudiart au sujet de l'état de l'art
- - Responsable : Thomas Kieffer
- - Délai :Vendredi 12/11/2021
- - Validé par : Julien De Toffoli

### *Prochaine Réunion : 14/11/2021*

- - Réfléchir aux besoins en bases de données, algorithmes, développement web relatifs aux fonctionnalités évoquées
- - Réfléchir aux fonctionnalités prioritaires de l'application

## Compte rendu 14 Novembre 2019

Nous sommes le 14 Novembre 2021, Julien, Pierre, Nathan et Thomas sont présents. L'objectif de cette réunion est de définir les besoins de notre application et de la priorisation des tâches pour être plus efficaces. La réunion a lieu sur Discord à 20h.

### *Décisions principales*

- - Mise en place d'un ordre des tâches à effectuer et conceptualisation de la partie offre/demande et aussi de la conception de profil
- - Consensus sur les différents algo et les informations dont on aura besoin pour obtenir les résultats escomptés

### *To Do List*

- — - Description : Aspect global du site WEB
  - - Responsable : Julien De Toffoli
  - - Délai : Pour la prochaine réunion (18/11/21)
  - - Validé par : Nathan Iori-Gingembre
- — - Description : Conception du tableau regroupant les besoins en algo observés pendant la réunion
  - - Responsable : Thomas Kieffer
  - - Délai : Pour la prochaine réunion (18/11/21)
  - - Validé par : Nathan Iori-Gingembre

### *Prochaine Réunion : 18/11/2021*

- - Penser aux façons de faciliter l'utilisation du site par les usagers

## Compte rendu 18 Novembre 2021

Nous sommes le 18 Novembre 2021, Julien, Pierre, Nathan et Thomas sont présents. L'objectif de cette réunion est de finaliser la conception du projet. La réunion a lieu sur Discord à 20h.

### *Décisions principales*

- - Validation des schémas de pages réalisés par Julien
- - Fournir l'application d'un algorithme de traitement

### *To Do List*

- - Description : Réfléchir à un algorithme de traitement que l'on pourrait mettre en place
- - Responsable : Tout le monde
- - Délai : 21/11/2021
- - Validé par : Tout le monde

### *Prochaine Réunion : après la validation du concept*

- - Faire le WBS
- - Commencer à distribuer les tâches à effectuer pour l'implémentation de l'application

## Compte rendu 20 Décembre 2021

Nous sommes le 20 Décembre 2021, Julien, Pierre, Nathan et Thomas sont présents. L'objectif de cette réunion est de discuter du WBS et de la répartition sous-jacente des responsabilités avec la RACI. La réunion a lieu sur Discord à 20h.

### *Décisions principales*

- - Etablir l'ensemble des URL, pour faciliter le développement ultérieur du fichier app.py
- - Organiser la répartition du travail de telle sorte que le développement de chaque page se fasse en parallèle de la création de requêtes dans la base de données et avec le routage correspondant
- - Etablir les responsabilités pour chaque lot de travail

### *To Do List*

- - Description : Réaliser l'ensemble des templates et des requêtes relatives au bon fonctionnement de chaque template
- - Responsable : Tout le monde
- - Délai : Lundi 27/12/2021
- - Validé par : Thomas Kieffer

### *Prochaine réunion : 24/11/2021*

- - Discuter de l'avancement de chacun.
- - Evoquer les éventuels problèmes rencontrés, essayer d'y remédier.

## Compte rendu 24 Décembre 2021

Nous sommes le 24 Décembre 2021, Julien, Nathan et Thomas sont présents. L'objectif de cette réunion est d'observer l'avancement mutuel et de tenter de résoudre les problèmes survenus. La réunion a lieu sur Discord à 11h.

### *Décisions principales*

- - Thomas : essayer d'utiliser Javascript pour les pages sondages
- - Julien : Mieux utiliser l'id dans l'URL
- - Nathan : pas de difficultés donc continue son développement

### *Prochaine réunion : 27/12/2021*

- - Observer l'avancement
- - Voir les difficultés

## Compte rendu 27 Décembre 2021

Nous sommes le 27 Décembre 2021, Julien, Pierre et Thomas sont présents. L'objectif de cette réunion est d'observer l'avancement et de résoudre les problèmes. La réunion a lieu sur Discord à partir de 18h.

### *Décisions principales*

- - Abandon Javascript pour sondages, utilisations d'entrées par l'utilisateur à la place
- - Résoudre les problèmes de certaines requêtes
- - Implémentation du système de filtre
- - Choix du nom de l'application HelpGo et couleurs du site

### *Prochaine réunion : 31/12/2021*

- - Observer l'avancement
- - Voir les difficultés



## Compte rendu 31 Décembre 2021

Nous sommes le 31/12/2021, Julien, Pierre, Nathan, Thomas sont présents. L'objectif de cette réunion est d'observer l'avancement et de résoudre les problèmes. La réunion a lieu sur Discord à partir de 16h30.

### *Décisions principales*

- - Modifications finales du layout et finalisation sondages
- - Rajouter des éléments à différentes tables pour compléter la cohérence des pages
- - Avancer sur le login
- - Finir tous les templates d'ici le 02/01/2022

### *Prochaine réunion : 02/01/2022*

- - Régler les derniers problèmes éventuels
- - Mise en lien de toutes les pages

## Compte rendu 02 Janvier 2022

Nous sommes le 2 Janvier 2022, Julien, Pierre, Nathan et Thomas sont présents. L'objectif de cette réunion est d'observer l'avancement et de résoudre les problèmes. La réunion a lieu sur Discord à partir de 18h30.

### *Problèmes :*

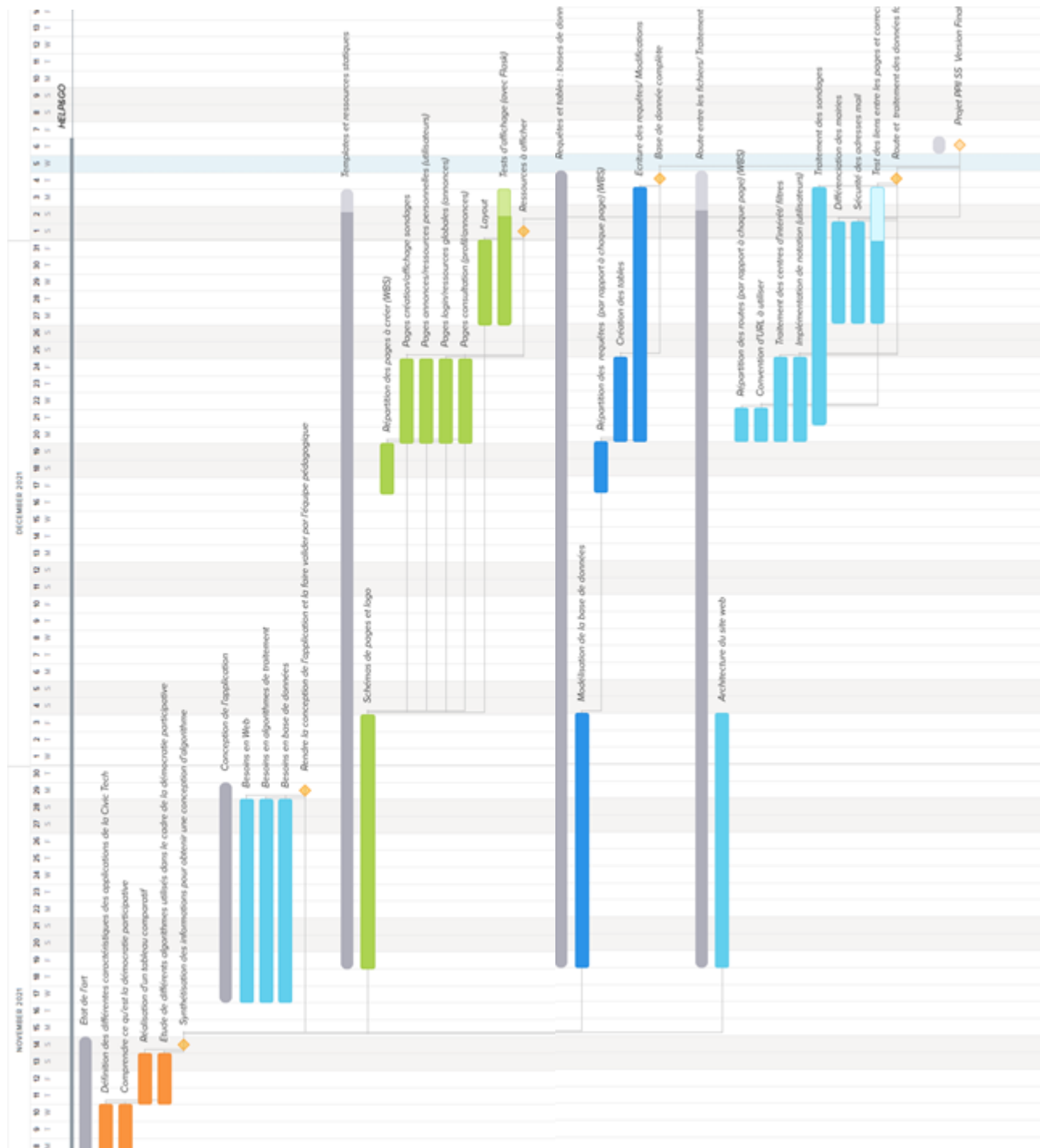
- - Problèmes de définitions des variables, sur leurs globalités et leurs contexte d'application.
- - Problème de réflexion pour afficher les postulants et les demandeurs aux annonces.
- - Problème dans des templates

## *Décisions :*

- - Mise en accord sur la possibilité de modifier le profil.
- Observation plus précise de ses pages et de ses templates.
- - Mise en rapport de tous les templates
- - Préparer le gestion de projet sous Latex



# Annexe 2 : Diagramme de Gantt



## Annexe 3 : Gestion des risques

